



# UNIVERSIDAD NACIONAL DE LOJA



**Facultad de Energía, Las Industrias y los  
Recursos Naturales No Renovables**

**Teoria de la Programación**

**Nombre:**

**RICARDO MATHIAS  
OCHOA ARMIJOS**

**Fecha:**

**24/11/2025**

**Docente:**

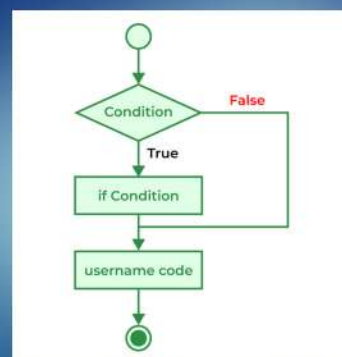
**Ing. Lissette Geoconda  
López Faicán**



# CONDICIONALES

## El if Simple

1



El if en C es la declaración de toma de decisiones más simple. Consiste en la condición de prueba y un bloque de código que se ejecuta si y sólo si la condición dada es verdadera. De lo contrario, se omite de la ejecución.

## Funcionamiento de la declaración if en C:

PASO 1: Cuando el control del programa llega a la declaración if, se evalúa la expresión de prueba.

PASO 2A: Si la condición es verdadera, se ejecutan las declaraciones dentro del bloque if.

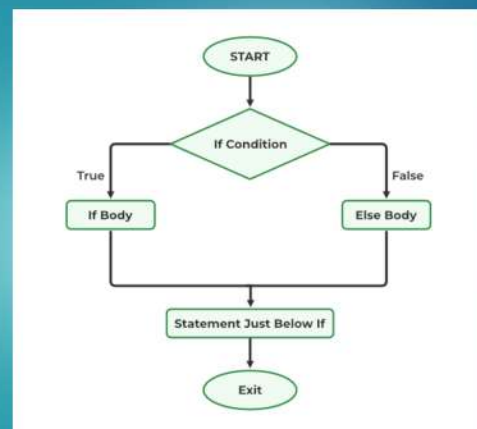
PASO 2B: Si la expresión es falsa, las declaraciones dentro del cuerpo if no se ejecutan.

PASO 3: El control del programa sale del bloque if y del código después de ejecutar el bloque if.

## El If Else

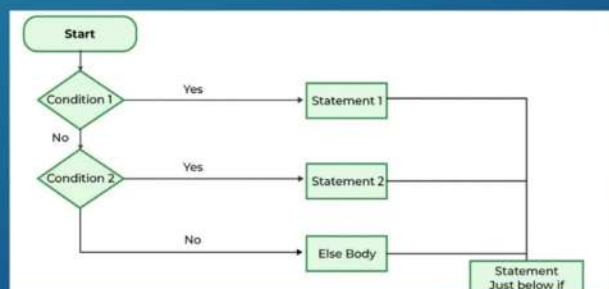
2

El If Else en C es una extensión del declaración if lo que no sólo permite al programa ejecutar un bloque de código si una condición es verdadera, sino también un bloque diferente si la condición es falsa.



```
C conodicional.c > main()
1  #include <stdio.h>
2
3  int main(){
4
5      int nota = 85;
6
7      printf("Ingrese su nota: ");
8      scanf("%i", &nota);
9
10     if (nota == 100) {
11         printf("Nota perfecta.\n");
12     } else if (nota >= 80) {
13         printf("Muy buena nota.\n");
14     }
15
16 }
```

- Si la condición es verdadera, entonces se ejecuta el código dentro del bloque if; de lo contrario, se ejecuta el código dentro del bloque else.
- Se supone que todos los valores distintos de cero y no nulos son verdaderos, y los valores cero o nulos se suponen falsos.
- El declaración if-else funciona comprobando la condición definida con la declaración if.
- Si la condición definida en el if La declaración es verdadera, entonces el código dentro de la if Se ejecuta el bloque y se omite el resto.



## El if-else if

3

• Se utiliza para probar una serie de condiciones secuencialmente, ejecutando el código para la primera condición verdadera. Una condición se verifica sólo si todas las anteriores son falsas.

• Una vez que una condición es verdadera, su bloque de código se ejecuta y la escalera finaliza.

```
C conodicional.c > main()
1  #include <stdio.h>
2
3  int main(){
4
5      int temperatura = 10;
6
7      printf("Ingrese la temperatura:");
8      scanf("%i", &temperatura);
9
10     if (temperatura > 30) {
11         printf("Hace calor.\n");
12     } else if (temperatura >= 20) {
13         printf("Clima agradable.\n");
14     } else {
15         printf("Hace frio.\n");
16     }
17
18 }
```





# Dado un número entero positivo que denota , haz lo siguiente:

Si ,  $1 \leq n \leq 9$  imprima la palabra inglesa en minúsculas correspondiente al número (por ejemplo, one para 1 , two 2 para 2, etc.).

Si ,  $n > 9$  imprimir Greater than 9.

Formato de entrada

La primera línea contiene un solo número entero, .

Restricciones

$1 \leq n \leq 10^9$

Formato de salida

Si ,  $1 \leq n \leq 9$  luego imprima la palabra inglesa en minúsculas correspondiente al número (por ejemplo, one para 1 , two para 2 , etc.); de lo contrario, imprimir Greater than 9 en cambio

```
C numeroIngles.c > main()
1  #include <stdio.h>
2
3  int main(){
4
5      int numero;
6      char *ingles;
7
8      printf("Ingrese un numero: ");
9      scanf("%i", &numero);
10
11     if ( 1 == numero)
12     {
13         ingles = "One";
14     }else if (2 == numero)
15     {
16         ingles = "Two";
17     }else if (3 == numero)
18     {
19         ingles = "Three";
20     }else if (4 == numero)
21     {
22         ingles = "Four";
23     }else if (5 == numero)
24     {
25         ingles = "Five";
26     }else if (6 == numero)
27     {
28         ingles = "Six";
29     }else if (7 == numero)
30     {
31         ingles = "Seven";
32     }else if (8 == numero)
33     {
34         ingles = "Eight";
35     }else if (9 == numero)
36     {
37         ingles = "Nine";
38     }else
39     {
40         printf("Greater than 9 ");
41         return 0;
42     }
43
44     printf("El numero %i en ingles es: %s ", numero, ingles);
45
46     return 0;
47 }
48 }
```

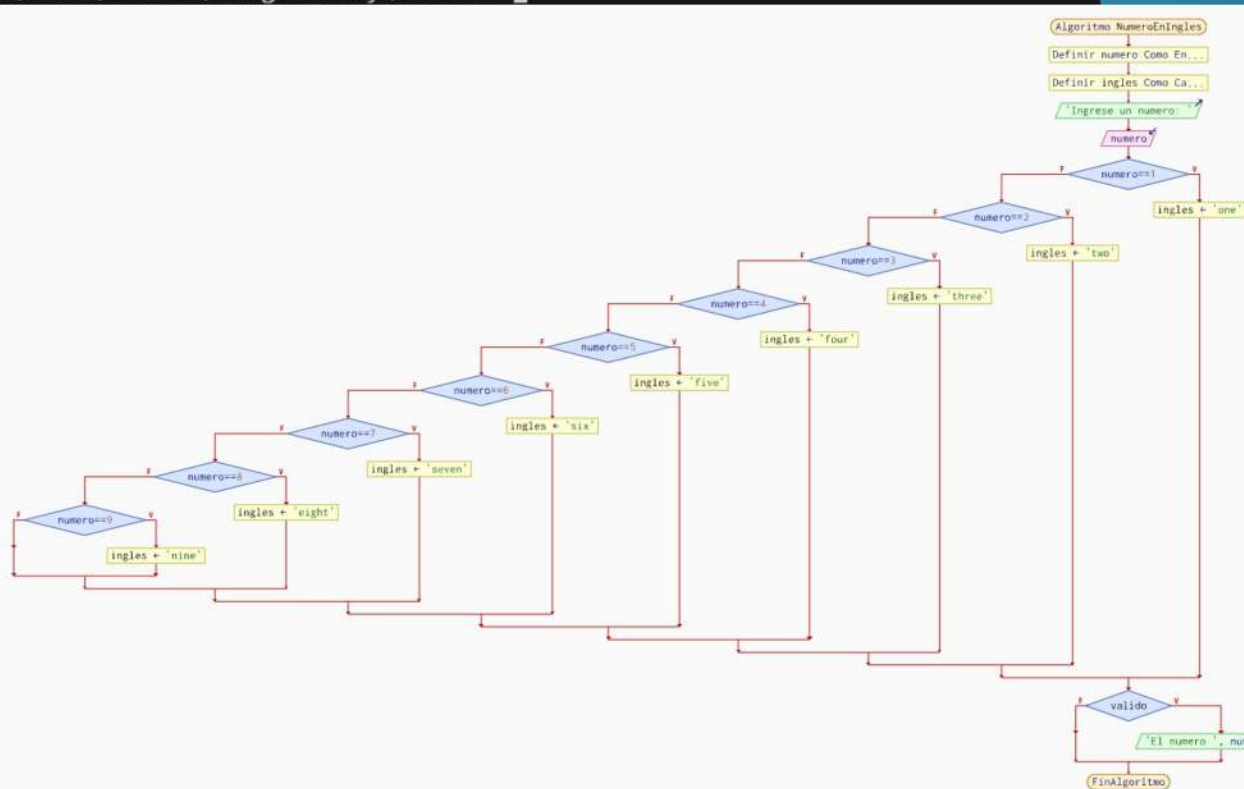
```
PS C:\Users\USUARIO\Codigos Ricky\Unida 2> gcc numeroIngles.c -o numeroIngles
```

```
PS C:\Users\USUARIO\Codigos Ricky\Unida 2> .\numeroIngles.exe
```

```
Ingrese un numero: 5
```

```
El numero 5 en ingles es: Five
```

```
PS C:\Users\USUARIO\Codigos Ricky\Unida 2> |
```



## Conclusiones

Las estructuras condicionales son importantes porque permiten que un programa tome decisiones según los datos que recibe. Sin estas condiciones, el programa solo haría lo mismo siempre, sin adaptarse a diferentes situaciones. Al usar "if", "else if" o "else", es posible controlar el camino que sigue el programa y lograr que responda de manera correcta a cada caso. Gracias a esto se pueden resolver problemas reales, porque la computadora puede elegir qué hacer en cada momento. Por esa razón, las estructuras condicionales son una parte esencial para construir soluciones en programación

## Bibliografía

[1] geeksforgeeks, «C if, else if ladder,» geeksforgeeks, Noida, 2022.

[2] geeksforgeeks, «C if else Statement,» geeksforgeeks, Noida, 2022.

[3] geeksforgeeks, «C - if Statement,» geeksforgeeks, Noida , 2022.